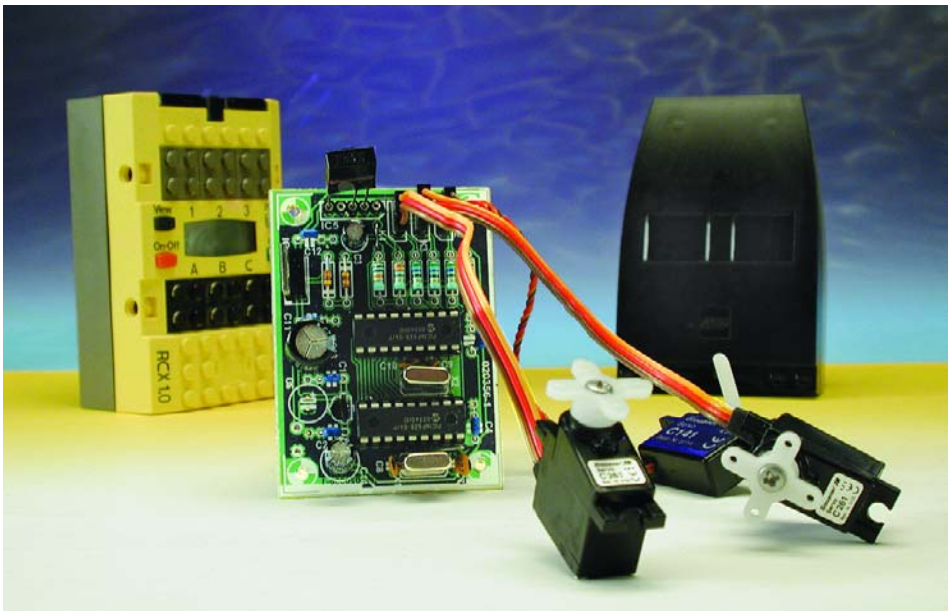
Interface Servomoteur IR pour RCX

Télécommande IR pour servo de modélisme compatible RCX Lego

Claude Baumann

L'un des seuls reproches que l'on puisse faire au système Mindstorms de Lego est qu'il lui manque des moteurs à positionnement tant soit peu précis. Cette constatation a incité l'auteur à développer une interface permettant le pilotage simultané de 3 servo-commandes de modélisme par le biais de l'interface IR du composant RCX (alias le bloc « jaune »).



Pour la plupart des projets de robots, les 3 types de moteurs 9 V de Lego Technique font parfaitement l'affaire. Cependant, dès lors que l'on cherche à réaliser des positionnements angulaires plus précis, on ne manquera pas de grincer des dents en constatant le jeu présenté par les engrenages; le besoin de solutions meilleures n'en deviendra que plus évident.

Les servo-commandes utilisées dans le monde du modélisme peuvent se targuer d'une précision bien supérieure. Cependant, on se trouve confronté alors à des problèmes non seulement de connexion mécanique mais aussi de pilotage par le biais du RCX qu'il n'est pas possible d'éliminer d'un coup de baguette magique.

Nous n'avons pas la prétention de vous apprendre que le RCX est le composant de pilotage central d'un système Mindstorms. Il s'agit en fait d'un petit micro-ordinateur intégré dans un bloc Lego de couleur jaune de dimensions supérieures à celles d'un bloc Lego standard.

Ralph Hempel, le créateur de pbForth, et Andreas Peter ont développé des modules auxiliaires qui permettent l'utilisation de 2 moteurs de servo avec RCX. On trouvera les descriptions correspondantes sur Internet aux adresses données en références [1] et [2]. Ralph Hempel a prévu, pour la commande de ses

Note : LEGO, Mindstorms, ROBOLAB, RCX et le logo LEGO sont des marques déposées du Lego Group.
Pour de plus amples informations, veuillez vous référer à : www.lego.com/eng/info/fairplay.asp.
Labview est une marque déposée de National Instruments Corporation

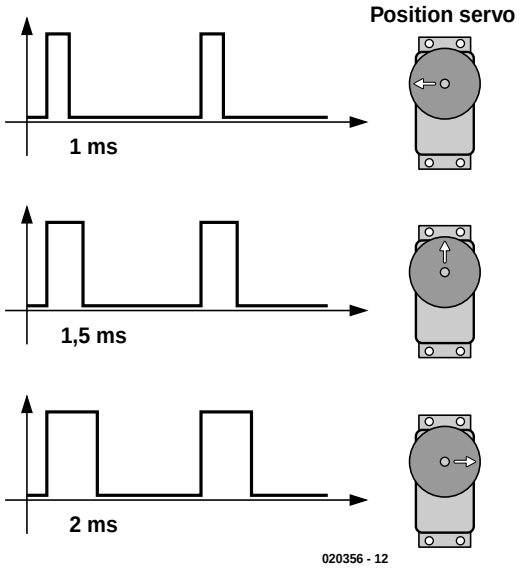


Figure 1. Principe de la commande d'une servo par le biais d'un signal impulsionnel dont la largeur d'impulsion représente la position.

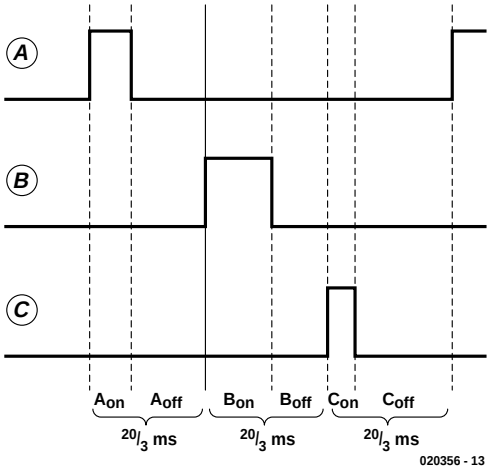


Figure 2. Chronodiagramme du signal de sortie du PIC utilisé pour la commande de 3 servos.

modules, des instructions de programmation spécifiques en pbForth, à savoir SERVO INIT et SERVO SET [3].
On pourra, pour attaquer des servo-commandes à base du progiciel 2.0 RCX standard (RCX-Standard-Firmware 2.0), mettre à contribution l'interface proposée ici ; elle permet la commande d'un maximum de 3 servos par l'intermédiaire de l'interface IR du RCX.

Commande de servo

La commande des servos utilisées en modélisme se fait par le biais d'une sorte de MLI (**M**odulation en **L**argeur d'**I**mpulsion = PWM pour *Pulse Width Modulation* en anglais). Le signal de commande est un signal rectangulaire ayant une fréquence de l'ordre de 50 Hz. Dans le cas présent la fréquence exacte ne joue aucun rôle dès l'instant où elle se situe entre 30 et 60 Hz. Le transfert de l'information fait uniquement appel à la largeur d'impulsion du signal rectangulaire, facteur qui varie entre 1 et 2 ms. Comme l'illustre le chrono-diagramme de la **figure 1**, la position centrale de la servo se situe à 1,5 ms. Au coeur de la servo, on a déplacement, parallèlement à l'axe de commande du disque, d'un potentiomètre de position qui fournit à l'électronique montée dans la servo l'information angulaire requise. Il est interdit de forcer la servo au-delà de ses limites d'un côté ou de l'autre en raison de la butée du potentiomètre, vu que sinon il y a risque de dommages irrémédiables. Ce pourrait être le cas en cas d'application d'impulsions dont la largeur sort du domaine autorisé.
Il faut, pour commander simultanément 3 moteurs de servo-commande, générer 3 signaux rectangulaires dont les largeurs d'impulsion doivent pouvoir être variées indépendamment l'une de l'autre.
Un coup d'oeil au schéma de la figure 3 montre que la source de ces signaux est IC3, un microcontrôleur du type PIC16F628. La structure du programme enfoui dans le PIC est telle qu'à 50 Hz, on a une succession chronologique des 3 impulsions de commande en respect de la relation suivante :

Aon + Aoff + Bon + Boff + Con + Coff = 1/50 s = 20 ms

Dans cette formule Aon, Bon et Con représentent les largeurs d'impulsion souhaitées. L'intervalle de temps de 20 ms dont nous disposons est subdivisé en 3 parties égales en respectant les conditions suivantes :

Aoff = 20/3 ms – Aon ; Boff = 20/3 ms – Bon ; Coff = 20/3 ms – Con

C'est cette chronologie que représente le diagramme impulsionnel de la **figure 2**. Grossièrement, le programme de commande et de génération d'impulsions prend (en pseudo-code) la forme suivante :

```
Label_1
      OutA on
      Wait Aon
      OutA off
      Wait Aoff

      OutB on
      Wait Bon
      OutB off
      Wait Boff

      OutC on
      Wait Con
      OutC off
      Wait Coff

      Goto Label_1
```

L'électronique de l'interface

L'alimentation de l'électronique de la **figure 3** se fera par le biais d'une pile compacte 9 V dédiée. La régulation de tension réalisée à l'aide d'une paire de régulateurs de tension, IC1 et IC2, garantit un fonctionnement sans problème sans chute de tension au cours du fonctionnement du moteur de servo. En cas de commande simultanée de 3 moteurs, le courant grimpe jusqu'à 500 mA environ. Pour cette raison, il ne saurait être question de brancher l'interface à une sortie dRCX qui n'est pas prévue à cette intention. À noter la

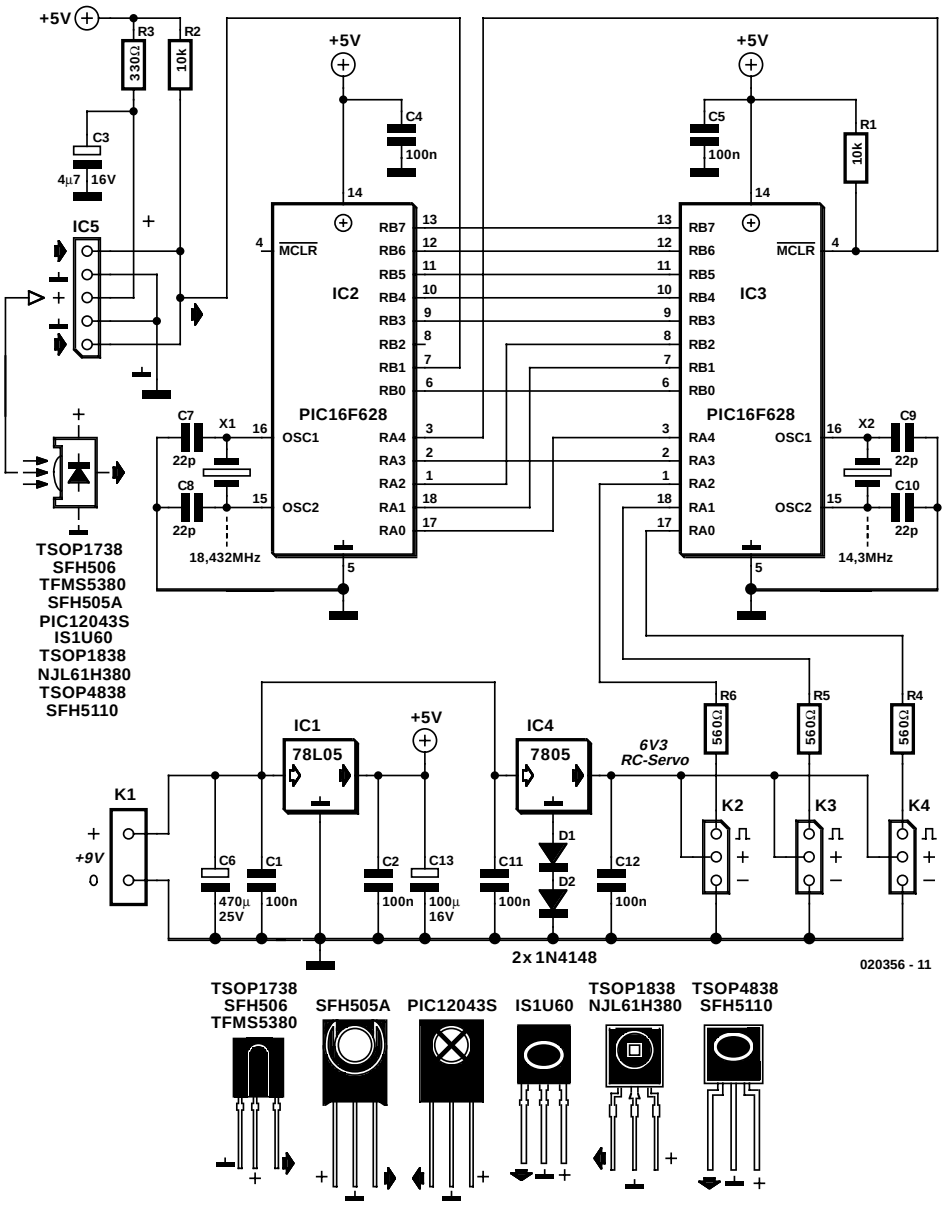


Figure 3. L'électronique se subdivise en 3 sous-ensembles : l'alimentation régulée (IC1 et IC4), le récepteur IR (TSOP1738 + IC2) et le générateur MLI (IC3).

disposition étrange des diodes D1 et D2. On a, sur chacune d'entre elles, une chute de tension de quelque 0,65 V, ce qui se traduit par un décalage du zéro de 1,3 V au niveau du 7805. De ce fait, la tension de sortie du régulateur IC4 est rehaussée de la même valeur et passe à la valeur de 6,3 V classique pour la commande des servo-commandes.

L'électronique de commande par infrarouge (IR) de moteur pas à pas proprement dite comprend le récepteur IR IC5 et une paire de microcontrôleurs du type PIC16F628 montés en succession. Le premier des 2 microcontrôleurs, IC2, transmet les données sérielles qui lui ont été fournies par le récepteur IR sous format parallèle au second contrôleur, IC3. Ce processus se fait par le biais d'un bus de données à 8 bits et d'une ligne de commande à 3 bits. On déduit de ces chiffres que la résolution maximale est de 256. L'intervalle de temps de 1 ms est subdivisé en 255 pas, ce qui se traduit, en fonction des caractéristiques de la servo, à une précision angulaire meilleure que 1 °. Il se peut que sur certains types de servos, cette résolution soit supérieure à la zone morte (*dead-band*, de 5 µs typique) de sorte qu'il peut arriver que le moteur de la servo ne réagisse pas à un pas individualisé (Note : les moteurs de servo ont besoin de la zone morte de manière à pouvoir, en cours de mouvement de rotation, rendre le plus rapidement possible le signal de commande entrant égal au signal de référence interne quelque peu en retard, sans qu'il ne se produise de battement du rotor aux alentours de l'angle de consigne).

Le second PIC, IC3, est cadencé à une fréquence d'horloge de 14,3 MHz dérivée du rapport durée d'impulsion servo/cycle PIC. Cette fréquence correspond à un train d'horloge de 69,9 ns. À l'image de son prédécesseur très apprécié, le 16F84, l'exécution de la plupart des instructions RISC se fait, dans le cas du 16F628, en 4 cycles d'horloge, y compris l'incrémentation du temporisateur. De ce fait, une instruction dure 279,9 ns. Le programme enfoui dans IC3 fait appel au temporisateur (*timer*) 16 bits TMR1. Le signal de servo le plus court, celui de 1,0 ms, correspond ainsi à 3 575 incrémentations de TMR1. À noter qu'il s'agit également là du paramétrage de départ. Il est suivi par une pause OFF de (20/3 – 1) ms qui correspond à 20 258 impulsions d'horloge. La durée d'impulsion exprimée en cycle de PIC répond à la formule suivante :

$t = a * x + b$, formule dans laquelle

$0 < x \leq 255$, 0 est réservé à des fins spécifiques

$a = (7\,150 - 3\,575)/(255-1) = 14{,}0$, Pente (*slope*)

$b = 3\,575$, Décalage du point zéro (*intercept*)

Ceci explique le choix de la fréquence d'horloge pour IC3 : le rapport 3 575/254 est presque entier. Le programme du PIC a en effet été doté d'une capacité de modification de la pente et du décalage du point zéro évoqués dans la formule quelques lignes plus haut, de manière à permettre un ajustement (*trim*) des servos. Il faut, dans le protocole de transmission, respecter certaines limites qui recommandent d'envoyer de données de 8 bits seulement. C'est pour cette raison que la pente et l'interception ne peuvent être modifiées que par nombre entier (l'interception par pas de 20 uniquement).

Tableau 1 Valeurs des lignes de commande

IC2 (UART)

A4	A3	A0	Valeur	Valeur PortA (A7-A5, A2, A1 masqué)	Signification
0	0	0	0	0	NOP
0	0	1	1	1	Charge A
0	1	0	2	8	Charge B
0	1	1	3	9	Set Slope
1	0	0	4	16	Charge C
1	0	1	5	17	Set Intercept
1	1	0	6	24	NOP
1	1	1	7	25	Reset

Tableau 3

Le set d'instructions de Robolab (Device Instruction Set)

Instruction	Numéro	Commande
1 (1)	0	Select channel A
2 (8)	0	Select channel B
4 (16)	0	Select channel C
1 (1)	1..255	Charge A
2 (8)	1..255	Charge B
4 (16)	1..255	Charge C
3 (9)	0..255	Set slope of selected channel (default 14)
5 (17)	0..5100 (0..255)	Set intercept (DIV 20) of selected channel (default 3575)
7 (25)	x	Reset all the slopes and intercepts to default

Transfert de données

Le transfert parallèle des données entre IC2 et IC3 se fait en 2 étapes. La mise en oeuvre (exceptionnelle) de 2 microcontrôleurs se justifie par le fait que l'on affaire à un certain nombre de processus à la chronologie extrêmement critique à exécuter. Il faut, pour commencer, que l'UART (*Universal Asynchronous Receiver/Transmitter* = Émetteur/Récepteur Asynchrone Universel) se trouve en permanence en mode réception, vu que l'interface n'a aucun moyen de savoir quand doit arriver une instruction IR. Il faut en outre que l'UART traite les données reçues conformément au protocole LEGO utilisé (code opératoire 0x14), ce qui se traduit, dans le cas d'un taux de transmission (*baudrate*) de 2 400 bauds, par une taille de bloc de 15 octets (bit de début, 8 bits de données, bit de parité, bit d'arrêt compris) et implique une durée de transfert de quelque 70 ms par instruction. Il faut en outre, en fonction des instructions entrant, générer 3 signaux MLI stables dont la tolérance doit être inférieure à la zone morte de la servo. La difficulté d'intégrer tous ces facteurs dans un programme tenant la route enfoui dans un unique PIC est sans rapport avec le coût d'un second 16F628.

Les broches RB1 et RB2 de IC2 sont configurées en UART et ne sont partant plus disponibles en tant que partie du bus de données parallèle. Elles sont remplacées par RA1 et RA2. Cette disposition requiert la mise en oeuvre d'au moins une ligne de commande vu qu'il n'est pas possible de positionner de façon synchrone les deux ports de IC2. En l'absence de ligne de commande, IC3 réagirait à l'apparition de valeurs intermédiaires parasites. À noter, au niveau des lignes de commande, que RA4 de IC2 requiert, étant une sortie à drain ouvert, une résistance de forçage au niveau haut (*pull up*) qui prend la forme de R1 (10 kΩ).
Les 2 circuits intégrés connaissent une configuration de base qui force, en interne, ~~MCLR~~ à VDD. De ce fait, la broche de IC3 est utilisable en entrée RA5. Le **tableau 1** récapitule les valeurs des lignes de commande et leur signification.

Tableau 2

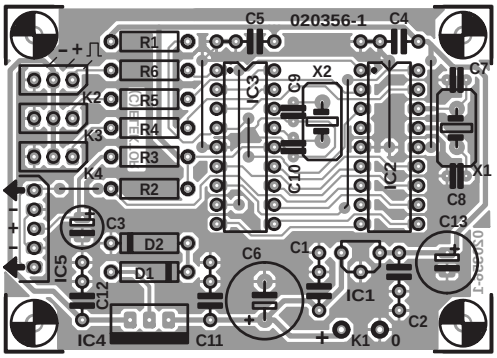
Données additionnelles du protocole LEGO pour le contrôle de la transmission

En-tête (pour tous les Opcodes)	**hex 55 FF 00**
Opcode	0x14 (=d20)
Opcode-Complement	0xFF-0x14 = 0xEB
Dummy	0x00
Dummy-Complement	0xFF
Dummy	0x00
Dummy-Complement	0xFF
Instruction	n
Instruction-Complement	not(n)
Data	x
Data-Complement	not(x)
Checksum	s
Checksum-Complement	not(s)

UART du RCX

On apprend, à l'adresse de LEGO Mindstorms SDK2 [4], que la version de progiciel Firmware 2.0 permet un accès utilisateur au canal d'émission de l'UART du RCX. Cette possibilité ouvre des perspectives de projets robotiques insoupçonnées. Il est possible ainsi, dans ces conditions, de faire programmer un RCX par un autre RCX, ou encore de faire en sorte que plusieurs modules RCX traitent des capteurs ou des valeurs de variables communs et ainsi de suite. Il suffit de respecter le protocole UART de LEGO et d'utiliser les codes opératoires convenables. Il faut cependant tenir compte du fait que tous les modules RCX sont adressés à l'intérieur du domaine de réception des instructions IR. D'après SDK cela peut avoir des « suites catastrophiques ». Nous avons, pour de très bonnes raisons, opté pour l'utilisation du code opératoire 0x14 (Servar ou Setv). Cette instruction n'est plus utilisée sous Robolab 2.5 vu que Firmware 2.0 a vu l'introduction d'une instruction SET universelle qui permet, en fonction du paramétrage, de déterminer des valeurs de capteurs (*clicks*) et autres variables.
Si l'on en croit la documentation technique, le code opératoire complet prend la forme suivante :

SETVAR: 0x14 variable, source, number (LO), number (HI)
(**Reply 0xE3**, il faut tenir compte du fait que le module RCX récepteur envoie, après vérification d'un code opératoire, une confirma-



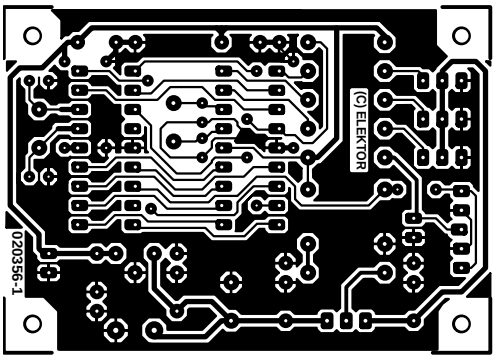


Figure 4. Un dessin de platine compact; il ne faudra pas oublier les ponts de câblage...

Liste des composants

Résistances :
R1,R2 = 10 kΩ
R3 = 330 Ω
R4 à R6 = 560 Ω

Condensateurs :
C1,C2,C4,C5,C11,C12 = 100 nF
C3 = 4µF7/16 V radial
C6 = 470 µF/16 V radial
C7 à C10 = 22 pF
C13 = 100 µF/16 V radial

Semi-conducteurs :
D1,D2 = 1N4148
IC1 = 78L05
IC2,IC3 = PIC16F628 (programmés, **EPS 020356-41** (IC2)* et **-42** (IC3)*
IC4 = 7805
IC5 = TSOP1738 (cf. texte)

Divers :
K1 = pile 9 V-compacte (cf. texte)
K2 à K4 = embase autosécable mâle à 1 rangée de 3 contacts
X1 = quartz 18,432 MHz
X2 = quartz 14,3 MHz (cf. texte)
Platine 020356-1*
Diskette EPS 020356-11*
*Les contrôleurs programmés, la platine et les disquettes sont disponibles auprès des adresses habituelles dont www.elektor.fr/pcbs/pcbs.htm) – pour le contenu de la disquette cf. L'encadré « Téléchargements pour ce projet »

tion de réception. Dans le cas qui nous intéresse, cela n'a pas la moindre importance. Cependant, il faudra en tout cas insérer une petite pause entre 2 émissions de codes opératoires vu que l'on risque sinon des interférences IR).

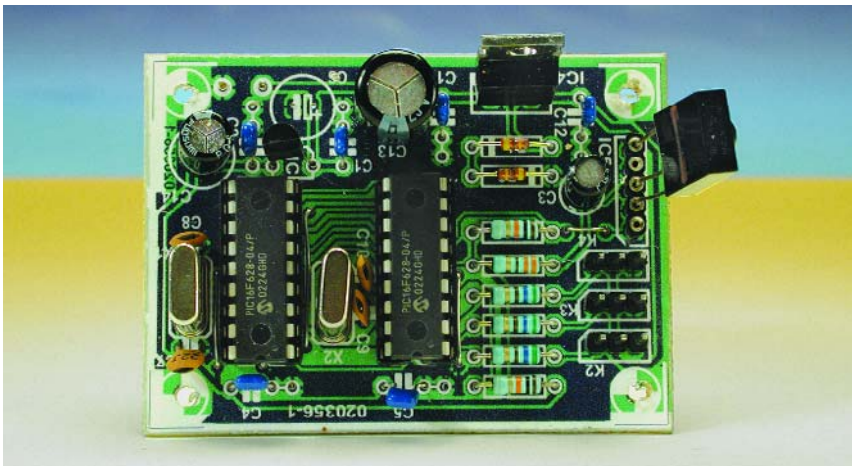
Cette instruction subit de petites modifications :

0x14 0 (dummy), 0 (dummy), instruction, data
(Les RCX concernés perdent malheureusement le contrôle de la variable 0)

L'ensemble du protocole LEGO requiert quelques données additionnelles pour le suivi de la transmission, informations récapitulées dans le **tableau 2**. Si, côté récepteur, on calcule les nombres complémentaires et la somme de vérification des octets reçus, et que l'on vérifie ensuite si l'octet suivant est bien le bon, il y a un contrôle sérieux et efficace de la qualité de la transmission des données. Bien que le RCX lui-même procède encore à un test de parité, cela n'est pas nécessaire au niveau du programme du PIC IC2 vu que l'UART 16F628 ne procède pas à un test de parité automatique. Pour le circuit intégré récepteur/démodulateur IR, l'auteur a opté pour un TSOP1738 de Vishay-Telefunken au prix très abordable.
La transmission IR LEGO se fait à 38 kHz (porteuse) et 2 400 bauds. Le récepteur IR sensible permet une portée pouvant aller jusqu'à 10 mètres si tant est que le RCX soit paramétré pour une (grande) liaison IR.
Le **tableau 3** passe en revue les instructions utilisées par les programmes de Robolab. Les valeurs réellement envoyées sont mises entre parenthèse. Comme nous le disions plus haut, il faut procéder avec précaution lors de la commande de débattement (*trim*). Pour éviter de faire sortir la servo de son domaine, avec comme conséquence un risque d'endommagement, il est fortement recommandé de ne travailler que par petits pas.

La réalisation

La construction de l'interface ne devrait pas poser de problème en dépit de la relative compacité de l'implantation des composants, ce grâce à la platine proposée en **figure 4**. La mise en place des ponts de câblage requiert un minimum d'attention, mais c'est là le prix à payer si l'on veut réaliser une platine simple face. Il y a 6 ponts au total, 3 d'entre eux se trouvent sous les circuits intégrés, l'un sous IC2 et deux sous IC3. On utilisera de préférence du conducteur isolé pour les dits ponts de câblage que l'on montera alors côté « pistes ». Une autre solution consiste à utiliser un fil de cuivre extrêmement fin que l'on glissera sous les supports avant la mise en place de ces derniers. En tout état de cause, on commencera toujours l'implantation des composants par la mise en place des ponts de câblage



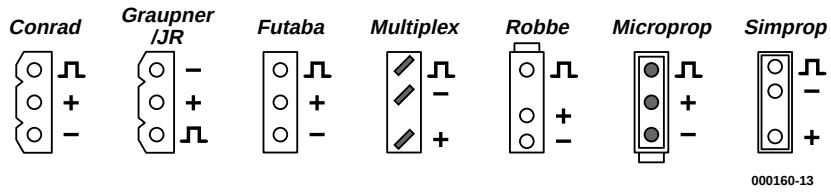


Figure 5. Brochage de différents modèles de servo-commandes.

avant de passer au soudage du reste des composants en commençant par les composants passifs. Si l'on veut être certain de réussir la réalisation il faut bien entendu non seulement implanter et souder correctement mais aussi respecter la polarité des composants en ayant une (condensateurs électrochimiques, diodes) et le positionnement des circuits intégrés. Au niveau du régulateur de tension IC4, la surface métallique doit être tournée vers l'extérieur de la platine.

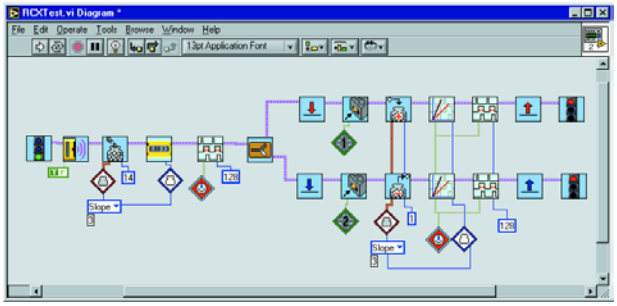


Figure 6. Un des programmes RCX à télécharger depuis www.elektor.fr. Ici, le positionnement de la servo se fait par le biais de capteurs sensitifs.

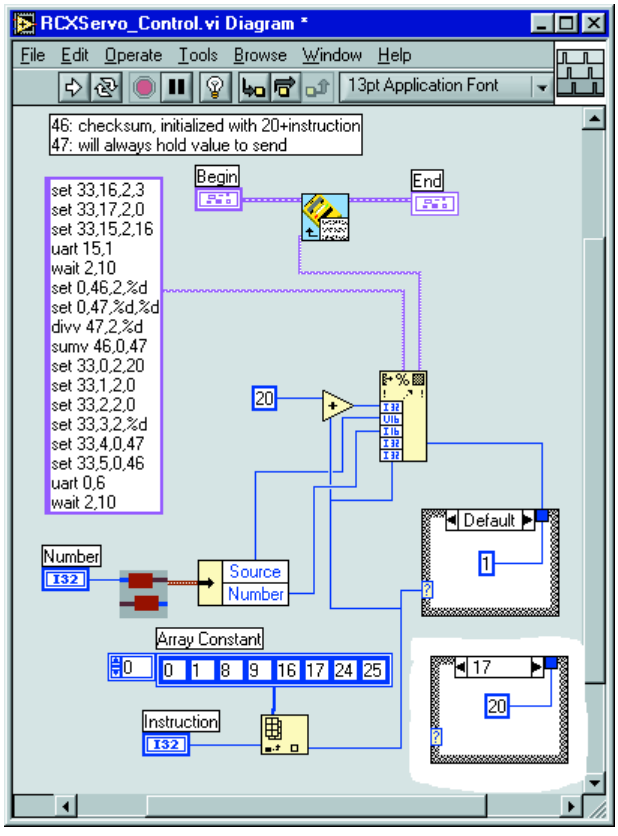


Figure 7. Le coeur du programme RCX : le mode émetteur de l'UART du RCX.

La connexion de la pile 9 V se fait par le biais de la paire de picots constituant K1. La consommation de courant du montage dépend principalement du nombre de servos et de la consommation de ces dernières. Si l'on dispose de suffisamment de place, il est préférable de remplacer la pile compacte de 9 V par un set de 6 piles LR6 (Mignon). La durée de vie d'une pile 9 V est en effet sensiblement moindre que celle d'une batterie de cellules LR6 (de 5 à 10 fois plus courte).

Il n'est pas impérativement nécessaire de monter le circuit intégré récepteur IC5 sur la platine. Il est fort bien possible de le décaler quelque peu et de le connecter au bornier baptisé IC5. Il est fort bien possible d'utiliser, au lieu du TSOP1738 de Vishay-Telefunken donné dans la liste des composants, d'autres récepteurs IR aux caractéristiques similaires. Nous vous proposons le brochage d'un certain nombre d'entre eux dans le schéma de la figure 4. Le type TSOP1736 est utilisable lui aussi (sa sensibilité est cependant moindre, sachant qu'il travaille sur 36 et non pas 38 kHz). Chacun de ces composants est tripode. Vous n'aurez pas manqué de noter que le bornier IC5 comporte 5 points de connexion; il a été cependant dessiné de façon à ce pour chacun des composants mentionnés, on utilise à chaque fois 3 broches juxtaposées ce qui facilite le soudage ou la connexion à distance.

Au niveau du quartz de 14,3 MHz aussi il existe une alternative. Si l'on ne devait pas disposer ou pouvoir mettre la main sur un quartz de cette fréquence on pourra le remplacer par un quartz plus facile à trouver, un 16 MHz par exemple. Les servos de radiocommande du commerce n'ont pas de problème avec la tolérance introduite par ce remplacement.

Les 3 servo-commandes viennent se brancher sur les embases K2, K3 et K4. À ce niveau aussi il faudra veiller à respecter le brochage. En effet, selon leur fabricant, le brochage du câble de connexion de la servo diffère. Nous vous proposons en **figure 5**, le brochage des câbles de connexion des servos les plus courantes.

Mode d'emploi

L'interface de commande de servo se laisse piloter sous la houlette de Robolab 2.5 si tant est que l'on soit en RCX-Firmware 2.0, tant par la LEGO-Tower (RS-232 et USB) que directement par le RCX. Il existe, pour ce faire, différents programmes disponibles au téléchargement (les adresses correspondantes sont mentionnées dans l'encadré « Télécharge-

Téléchargements

Pour ce projet :

– Dessin de la platine au format .pdf à télécharger sous la dénomination **EPS 020356-1** sous www.elektor.fr/dl/dl.htm
– Programme RCX pour la commande de servos, code-source et .hex des 2 PIC sous la dénomination **EPS 020356-11** au même endroit
– L'auteur met à disposition la totalité de sa documentation, avec images au téléchargement sous la forme d'un fichier à l'adresse : www.convict.lu/IR_Servo_control.zip

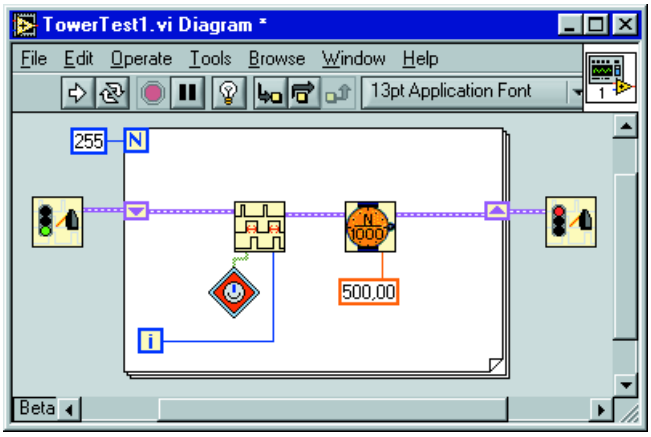


Figure 8. Ce programme simple tourne sur PC; il pilote les servos par le biais de la LEGO-Tower.

ments pour ce projet »); permettez-nous de vous en présenter certains succinctement.
On découvre en **figure 6** un programme de téléchargement à 2 tâches parallèles. Dans cet exemple, le RCX augmentera ou diminuera l'angle de consigne de la servo rouge, en prenant comme référence la position médiane de la servo, en fonction du capteur sensitif qui aura été touché. Les couleurs rouge, bleu et jaune sont dérivées des codes de couleur classiques de ROBOLAB. La seconde icône de programme paramètre le RCX pour une distance IR longue. L'affichage indique la valeur de pente (*slope*) momentanée. Il est possible, de la même manière, d'ajuster la valeur d'interception. Des modifications du paramètre de pente augmentent ou diminuent le débattement de la servo. Des modifications du paramètre interception n'ont pas d'autre effet que de déplacer le point zéro de la servo (le neutre).
Comme à l'accoutumée sous ROBOLAB/LABVIEW, les parties de programmes sont attribuées graphiquement à de nouvelles icônes. De cette façon, mêmes les programmes les plus complexes restent lisibles. La **figure 7** montre le coeur du programme RCX : il s'agit de l'application émission de l'UART du RCX. On trouvera de plus amples informations sur la mise en oeuvre de l'UART dans la partie documentation après installation du programme Lego-Mindstorms-SDK [4].
Le programme simple représenté en **figure 8** tourne sur PC. Le moteur de la servo rouge tourne lentement sur son domaine de déplacement. À noter la boucle FOR de LABVIEW avec registre à décalage. Par les lignes pointillées circulent des données qui s'ajoutent lors de chaque itération du registre à décalage. Cela permet, par exemple, un suivi très efficace du programme dans l'attente d'une erreur potentielle.
Les 2 icônes extérieures servent, respectivement, à l'initialisation et à la clôture de la liaison vers la LEGO-Tower. Le programme se met en quête de la liaison avec la Tower établie sous ROBOLAB. On ne pourra pas utiliser pour cela le mode classique ROBOLAB-RCX Direct, vu que celui attend toujours, côté récepteur, une confirmation de réception de la part du RCX. Dans le cas présent, on se trouve en présence d'une émission unilatérale.
La **figure 9** montre, pour terminer, le panneau principal du programme RCServo-Manager dont la mise en oeuvre est extrêmement simple. Le débattement se fait par le biais de Slope et Intercept. La Tower n'émet qu'en cas de modification de l'une des valeurs. Si ces changements se font à courir, la valeur RATE détermine le taux de cadencement.

(020356)

[1] www.inchlab.net/2servo interface.htm
[2] www.hempeldesigngroup.com/lego/servos/
[3] Dave Baum, Michael Gasperi, Ralph Hempel, Luis Villa: "Extreme Mindstorms, An Advanced Guide to LEGO MINDSTORMS", Page 313.
[4] www.ceeo.tufts.edu/robolabatceeo/Resources/documentation/L EGOMindStormsSDK.zip

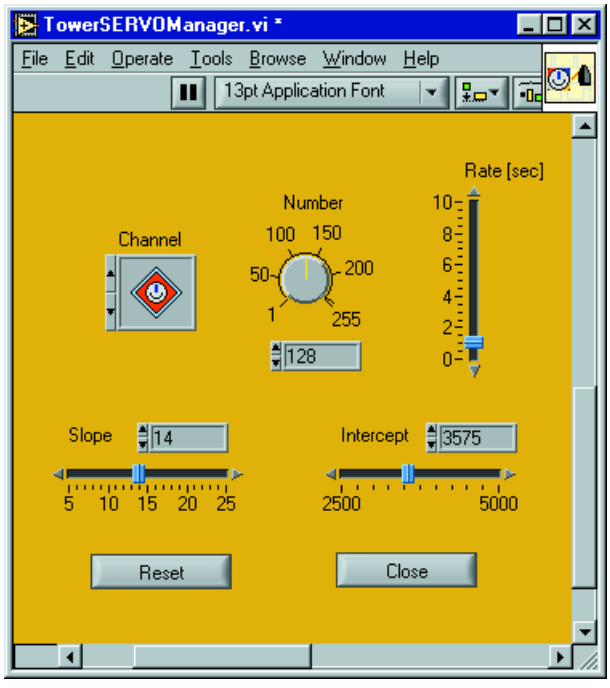


Figure 9. L'environnement du RCServo-Managers est de mise en oeuvre simple et confortable.